iFiG: Individually Fair Multi-view Graph Clustering

Yian Wang[1], Jian Kang[1], Yinglong Xia[2], Jiebo Luo[3], and Hanghang Tong[1]

[1]University of Illinois at Urbana-Champaign, {yian3, jiank2, htong}@illinois.edu
[2]Meta, yxia@fb.com
[3]University of Rochester, jluo@cs.rochester.edu

***Abstract*—In a connected world, fair graph learning is becoming increasingly important because of the growing concerns about bias. Yet, the vast majority of existing works assume that the input graph comes from a single view while ignoring the multi-view essence of graphs. Generally speaking, the bias in graph mining is often rooted in the input graph and is further introduced or even amplified by the graph mining model. It thus poses critical research questions regarding the intrinsic relationships of fairness on different views and the possibility of mitigating bias on multiple views simultaneously. To answer these questions, in this paper, we explore individual fairness in multi-view graph mining. We first demonstrate the necessity of fair multi-view graph learning. Building upon the optimization perspective of fair single-view graph mining, we then formulate our problem as a linear weighted optimization problem. In order to figure out the weight of each view, we resort to the minimax Pareto fairness, which is closely related to the Rawlsian difference principle, and propose an effective solver named iFiG that minimizes the utility loss while promoting individual fairness for each view with two different instantiations. The extensive experiments that we conduct in the application of multi-view spectral clustering and INFORM post-processing demonstrate the efficacy of our proposed method in individual bias mitigation.**

***Index Terms*—Clustering, individual fairness, multi-objective optimization**

I. INTRODUCTION

A multi-view graph comprises multiple single-view graphs with the same set of nodes but different types of edges. In many real-world applications, graphs are often collected from multiple sources, forming multi-view graphs. For example, users could have accounts on numerous social platforms like Facebook and Twitter; the infrastructure network of cities exhibits different topologies considering different types of infrastructures (e.g., power grid, road network). Up to now, researchers have proposed a variety of multi-view graph mining models, including clustering [1], embedding [2], and graph neural networks [3].

Despite many efforts in developing mining models with optimal utility (e.g., classification accuracy), the fairness aspect of multi-view graph mining is often understated. Take peer-to-peer (P2P) lending as an example, due to a lack of financial history of applicants on P2P lending platforms, existing works utilize social networks to improve the predictive accuracy [4]. Based on that, if the P2P lending service providers apply a prediction model to multiple social platforms with the same set of nodes for loan risk prediction, it is possible that two similar loans are both of low risk in one view but receive different classification results in another view. Instead of finding the optimal mining results for multi-view graphs, several key questions related to algorithmic fairness need to be answered: *Would ensuring fairness on one view unintentionally amplify the biases on another? If so, how can we mitigate the bias in multiple views simultaneously?*

The study of algorithmic fairness on graphs has attracted much attention, in which a majority of existing works [5]–[7] are on single-view graphs while ignoring the multi-view nature of graphs in many applications. To date, sparse efforts of fair non-single-view graph mining [8], [9] only ensure group fairness on heterogeneous graphs. Nevertheless, none of the existing works on fair graph mining is designed for multi-view graphs, nor is there work on individual fairness.

Different from group fairness, individual fairness follows a general principle of making similar individuals have similar outcomes, which has not yet been well studied for multi-view graph mining. Analogous to the general principle of individual fairness, Kang et al. [6] study individual fairness when mining single-view graphs by ensuring similar mining results for similar nodes (individuals). Specifically, individual fairness is ensured through Laplacian regularization on the pairwise node similarity matrix of a single-view graph (i.e., one view in a multi-view graph). However, simply ensuring individual fairness on one view might overlook the view heterogeneity among multiple views, which in turn could find unfair mining results for another view. Figure 1 provides an illustrative example where ensuring individual fairness on one view fails to ensure individual fairness on another view.

In this paper, we study individual fairness in multi-view graph mining (iFiG problem), which aims to ensure individual fairness on multiple views simultaneously. Our work not only defines fairness for multi-view graph mining but also provides a natural solution to ensure fairness in multi-view graph mining without sensitive attribute(s). We first formulate iFiG problem as a regularized optimization problem. To search for the regularization parameters that help balance the trade-off between utility and fairness, we develop its connection to finding the Pareto front of a multi-objective optimization (MOO) problem. To find the Pareto front with the best trade-off, we propose a generic algorithmic framework that leverages the Rawlsian difference principle to efficiently find the minimax

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

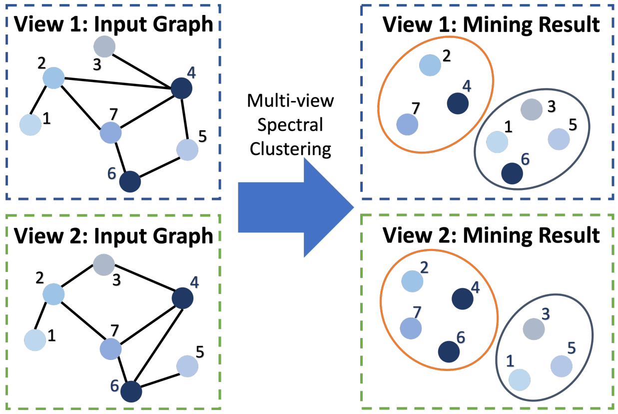

Figure 1: An illustrative example of peer-to-peer (P2P) lending on a two-view social network. Nodes represent borrowers and lenders, edges represent users' social networks on two different social platforms. Similar nodes in different views have the same transparency, meaning node 4 and node 6 have similar features. In View 1, two similar individuals are clustered into different groups because node 4 is highly connected and thus more likely to obtain higher social credit. In View 2, two similar individuals are clustered into the same group.

fair graph mining results.

The main contributions of the paper are as follows.

- **Problem Definition.** To our best knowledge, we are the first to ensure individual fairness in multi-view graph mining (iFiG problem).
- **Algorithmic Instantiation.** We propose an effective solver to find the Pareto front of the iFiG problem and instantiate it by multi-view spectral clustering and InForm post-processing.
- **Evaluations.** We evaluate our proposed method on a synthetic dataset and a variety of real-world datasets. Experimental results demonstrate the effectiveness of our methods in debiasing multi-view spectral clustering.

II. Preliminary

In this section, we first present the key symbols of this paper in Table I. After that, we briefly review multi-view graph mining and individual fairness on a single-view graph. Finally, we introduce the formal definition of individual fairness in multi-view graph mining.

Notations. We use bold upper-case letters for matrices (e.g., $\mathbf{A}$), bold lower-case letters for vectors (e.g., $\mathbf{x}$) and calligraphic letters for sets (e.g., $\mathcal{D}$). In addition, we use superscript ^{T} to represent matrix transpose (i.e., the transpose of $\mathbf{A}$ is $\mathbf{A}^T$).

A. Multi-view Graph Mining

We denote a multi-view graph with M views as a set of single-view graphs $\mathcal{G}^{\text{multi}} = \{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, where the i-th single-view graph in $\mathcal{G}^{\text{multi}}$ is $\mathcal{G}_i = \{\mathcal{N}, \mathbf{A}_i\}, \forall i \in \{1, \ldots, M\}$. For $\mathcal{G}_i$, $\mathcal{N}$ is the universal node set shared among all views in $\mathcal{G}^{\text{multi}}$, $\mathbf{A}_i$ is the view-specific adjacency matrix for i-th view. Many multi-view graph mining algorithms can be viewed as optimization problems that capture different relationships among views. From an optimization perspective, a multi-view graph mining model aims to optimize a task-specific loss function $r(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$ where $\mathcal{G}^{\text{multi}}$ is the input multi-view graph, $\mathbf{Y}$ is the mining results, and θ is the set of model parameters. Take the multi-view spectral clustering [10] as an example, whose goal is to learn an optimal universal cluster membership matrix that finds k high-quality clusters on each view and is close to the optimal cluster membership matrix concerning the corresponding view. It regards the mining results of each view as a subspace and aims to find the most representative subspace (i.e., the optimal multi-view graph mining results) to be close to any i-th individual subspace $\mathbf{Y}_i, \forall i \in \{1, \ldots, M\}$. Naturally, it is formulated as the following optimization problem

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \sum_{i=1}^{M} \text{tr}(\mathbf{Y}^T \mathbf{L}_i \mathbf{Y}) - \lambda \sum_{i=1}^{M} \text{tr}(\mathbf{Y}^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{Y}) \\ \text{s.t.} \quad & \mathbf{Y}^T \mathbf{Y} = \mathbf{I} \end{aligned} \tag{1}$$

where λ is the regularization hyperparameter and $\mathbf{Y}_i \in \mathbb{R}^{n \times k}$ contains the eigenvectors associated with k smallest eigenvalues of the graph Laplacian $\mathbf{L}_i$ of the i-th view. In Eq. (1), the first term $\sum_{i=1}^{M} \text{tr}(\mathbf{Y}^T \mathbf{L}_i \mathbf{Y})$ ensures that $\mathbf{Y} \in \mathbb{R}^{n \times k}$ finds the high-quality clusters on each view, and the second term $\sum_{i=1}^{M} \text{tr}(\mathbf{Y}^T \mathbf{Y}_i \mathbf{Y}_i^T \mathbf{Y})$ ensures the mining results $\mathbf{Y}$ to be close to the optimal cluster membership matrix $\mathbf{Y}_i, \forall i \in \{1, \ldots, M\}$ in each view. Analogous to single-view spectral clustering, the solution to Eq. (1) are essentially the eigenvectors corresponding to the first k smallest eigenvalues of the modified graph Laplacian $\mathbf{L}_{\text{mod}} = \sum_{i=1}^{M} \mathbf{L}_i - \lambda \mathbf{Y}_i \mathbf{Y}_i^T$.

B. Individual Fairness on a Single-view Graph

The general principle of individual fairness states that *any two similar individuals should be treated similarly* [11]. Mathematically, given a mining results $\mathbf{Y}$, it is often formulated as a Lipschitz inequality $d_1(\mathbf{Y}[i,:], \mathbf{Y}[j,:]) \leq \epsilon d_2(i,j)$ where $d_2(i,j)$ measures the distance between two data points i and j, $d_1(\mathbf{Y}[i,:], \mathbf{Y}[j,:])$ measures the distance between their corresponding mining results $\mathbf{Y}[i,:]$ and $\mathbf{Y}[j,:]$, and ϵ is the Lipschitz constant. Building upon it, Kang et al. [6] define

Table I: Table of symbols.

Symbol	Description
$\mathbf{A}$	a matrix
$\mathbf{A}^T$	transpose matrix of $\mathbf{A}$
$\mathbf{L}_{\mathbf{A}}$	Laplacian matrix of $\mathbf{A}$
$\mathbf{u}$	a vector
M	number of views in a graph
$\mathcal{V}$	view sets
$\mathcal{N}$	node set
$\mathbf{S}$	node similarity matrix
$\mathbf{Y}$	graph mining results
r	loss function
k	number of clusters
n	number of nodes
θ	a set of parameters

d_1 as the squared Frobenius norm such that $d_1(\mathbf{Y}[i,:], \mathbf{Y}[j,:]) = \|\mathbf{Y}[i;] - \mathbf{Y}[j;]\|_F^2$ and d_2 as the reciprocal of the corresponding entry in an oracle node similarity matrix $\mathbf{S}$ (i.e., $d_2(i,j) = \frac{1}{\mathbf{S}[i,j]}$). When the Lipschitz inequality holds for all pairs of nodes, the overall individual bias can be defined as $Bias(\mathbf{Y}, \mathbf{S}) = \text{tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$ where $\mathbf{L_S}$ is the graph Laplacian of node similarity matrix $\mathbf{S}$. As such, ensuring individual fairness on a single-view graph mining model is equivalent to the following optimization problem.

$$\min_{\mathbf{Y}} r(\mathcal{G}, \mathbf{Y}, \theta) + \alpha \text{tr}\left(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y}\right) \quad \text{s.t.} \mathbf{Y}^T\mathbf{Y} = \mathbf{I} \tag{2}$$

C. Problem Definition

Given a multi-view graph $\mathcal{G}^{\text{multi}}$ and a multi-view graph mining model that minimizes a task-specific loss $r(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$, our goal is to learn an optimal mining result by minimizing the task-specific loss while ensuring individual fairness on all views. Specifically, to achieve this goal, an individual fairness loss is incurred for each view. Formally, we define the problem of individual fairness in multi-view graph mining as follows.

Problem 1. *iFiG: Individual Fairness on Multi-view Graph Mining*

Input: (1) An undirected multi-view graph with M views $\mathcal{G}^{\text{multi}} = \cup_{i=1}^{M}\{\mathcal{G}_i\}$; (2) a non-negative symmetric node-node similarity matrix $\mathbf{S}_i$ for each View i ; (3) a multi-view graph mining model that minimizes a task-specific loss function $r(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$ with θ as the set of model parameters.

Output: A multi-view graph mining result $\mathbf{Y}^*$ that (1) minimizes the task-specific loss function $\mathbf{r}(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$ and (2) promotes individual fairness.

Problem 1 can be naturally formulated as the following optimization problem

$$\min_{\mathbf{Y}} \mathbf{r}(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta) + \sum_{i=1}^{M} \alpha_i \text{tr}(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}i} \mathbf{Y}) \tag{3}$$

where $\mathbf{r}(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$ is the task-specific loss function for multi-view graph mining, $\mathbf{L}_{\mathbf{S}_i}$ is the view-specific Laplacian matrix of similarity matrix $\mathbf{S}_i$ and α_i is the view-specific weighted parameter for fairness constraints.

It is intuitive to formulate Problem 1 as a regularized optimization problem in Eq. (3). Nonetheless, there are three critical questions that we need to address, including

- **(Q1)** Why do we have to combine the cost of each view together instead of applying a single-view graph mining algorithm to each view separately?
- **(Q2)** Compared with the single-view fair graph mining method (e.g., Eq. (2)), what is the extra complexity in terms of the optimization algorithm?
- **(Q3)** How can we automatically determine the weights (e.g., α_i) of different views?

III. Methodology

In this section, we provide our answers to the three key questions we pose in Section II-C with multi-view spectral clustering as an instantiation.[1] In particular, we first (Q1) show that ensuring fairness on one view may unintentionally amplify the bias on another. Then, we (Q2) formulate the problem of ensuring individual fairness on a multi-view graph as a linear weighted optimization problem. Based on that, we (Q3) propose an effective solver named iFiG and instantiate it by multi-view spectral clustering and InFoRM post-processing [6]. Afterward, we demonstrate that the rationality of using minimax lies in the Rawlsian principle.

A. Necessity of Individual Fairness on Multi-view Graphs

For a multi-view graph, enforcing fairness in one view can increase the bias in another view. Denote $\tilde{r}_{b2}$ as the bias of View 2 when enforcing fairness on it and r_{b2} the bias of View 2 when only enforcing fairness on other view(s).

We first present Lemma 1, which theoretically analyzes the possibility of bias amplification on one view when enforcing individual fairness on another view.

Lemma 1. *For a multi-view graph with M views, we denote the pairwise node similarity matrix $\mathbf{S}_i$ and its graph Laplacian as $\mathbf{L}_{\mathbf{S}_i}$. Then only minimizing the bias of View i (i.e., tr$(\mathbf{U}^T \mathbf{L}_{\mathbf{S}_i} \mathbf{U})$) can increase the bias of View j (i.e., tr$(\mathbf{U}^T \mathbf{L}_{\mathbf{S}_j} \mathbf{U})$), i.e., the enforcement of fairness in one view will hurt the fairness of other views.*

Proof: Generally, the loss function of a multi-view graph clustering algorithm for an M-view graph is

$$\text{tr}\left(\mathbf{Y}^T \left(\sum_{i=1}^{M} \mathbf{L}_i - \lambda \sum_{i=1}^{M} \mathbf{Y}_i \mathbf{Y}_i^T\right) \mathbf{Y}\right) \tag{4}$$

Without loss of generality, we consider a 2-view graph. Note that the proof can be extended to the case of multi-view graphs. Then, for a 2-view graph, the loss function of multi-view spectral clustering is

$$r = \text{tr}(\mathbf{Y}^T(\mathbf{L}_1 + \mathbf{L}_2 - \lambda \mathbf{Y}_1 \mathbf{Y}_1^T - \lambda \mathbf{Y}_2 \mathbf{Y}_2^T)\mathbf{Y}) \tag{5}$$

And the corresponding bias of View 2 is

$$r_{b_2} = \text{tr}(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}_2} \mathbf{Y}) = \sum_{j=1}^{k} \mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2} \mathbf{y}_j \tag{6}$$

where $\mathbf{y}_j$ denotes the j-th column of matrix $\mathbf{Y}$, i.e. the j-th eigenvector of matrix $\mathbf{Y}$. If we only enforce fairness on View 1 instead of View 2, the loss function should be

$$\tilde{r} = \text{tr}(\tilde{\mathbf{Y}}^T(\mathbf{L}_1 + \mathbf{L}_2 + \alpha_1 \mathbf{L}_{\mathbf{S}_1} - \lambda \mathbf{Y}_1 \mathbf{Y}_1^T - \lambda \mathbf{Y}_2 \mathbf{Y}_2^T)\tilde{\mathbf{Y}}) \tag{7}$$

Then the bias of View 2 can be written as

$$\tilde{r}_{b_2} = \text{tr}(\tilde{\mathbf{Y}}^T \mathbf{L}_{\mathbf{S}_2} \tilde{\mathbf{Y}}) = \sum_{j=1}^{k} \tilde{\mathbf{y}}_j^T \mathbf{L}_{\mathbf{S}_2} \tilde{\mathbf{y}}_j \tag{8}$$

By matrix perturbation theory [12], with the perturbation to a matrix $\Delta\mathbf{L} = \tilde{\mathbf{L}} - \mathbf{L}$, without loss of generality, we consider

[1] Note that our analysis could be naturally generalized to other multi-view graph mining settings.

$\mathbf{L}_{\mathbf{S}_1}$ as $\Delta\mathbf{L}$. Then the new eigenvalues and eigenvectors can be written as

$$\begin{aligned}
\tilde{\Lambda}_j &= \Lambda_j + \mathbf{y}_j^T \Delta\mathbf{L}\mathbf{y}_j \\
\tilde{\mathbf{y}}_j &= \mathbf{y}_j + \sum_{l=1, l\neq j} \frac{\mathbf{y}_l^T \Delta\mathbf{L}\mathbf{y}_j}{\Lambda_j - \Lambda_l}\mathbf{y}_j = \mathbf{y}_j + \Delta\mathbf{y}_j
\end{aligned} \tag{9}$$

Then, we get

$$\begin{aligned}
\tilde{r}_{b_2} &= \sum_{j=1}^{k} \tilde{\mathbf{y}}_j^T \mathbf{L}_{\mathbf{S}_2}\tilde{\mathbf{y}}_j = \sum_{j=1}^{k} (\mathbf{y}_j + \Delta\mathbf{y}_j)^T \mathbf{L}_{\mathbf{S}_2}(\mathbf{y}_j + \Delta\mathbf{y}_j) \\
&= \sum_{j=1}^{k} \mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j + \mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\Delta\mathbf{y}_j \\
&\quad + \Delta\mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j + \Delta\mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\Delta\mathbf{y}_j \\
&= r_{b_2} + \sum_{j=1}^{k} [\mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}(\sum_{l=1, l\neq j} \frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j}{\Lambda_j - \Lambda_l}\mathbf{y}_l) \\
&\quad + (\sum_{l=1, l\neq j} \frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j}{\Lambda_j - \Lambda_l}\mathbf{y}_l)^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j + \Delta\mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\Delta\mathbf{y}_j] \\
&= r_{b_2} + 2\sum_{j=1}^{k} \sum_{l=1, l\neq j} (\frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j}{\Lambda_j - \Lambda_l}\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j) \\
&+ \sum_{j=1}^{k} \Delta\mathbf{y}_j^T \mathbf{L}_{\mathbf{S}_2}\Delta\mathbf{y}_j \\
&= r_{b_2} + 2\sum_{j=1}^{k} \sum_{l=1, l\neq j} \frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j \mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j}{\Lambda_j - \Lambda_l} + O(\Delta\mathbf{y}^2) \\
&\approx r_{b_2} + 2\sum_{j=1}^{k} [\sum_{l<j} \frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j \mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j}{\Lambda_j - \Lambda_l} \\
&\qquad + \sum_{l>j} \frac{\mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_1}\mathbf{y}_j \mathbf{y}_l^T \mathbf{L}_{\mathbf{S}_2}\mathbf{y}_j}{\Lambda_j - \Lambda_l}]
\end{aligned} \tag{10}$$

Since the eigenvalues are sorted in the ascending order, $\Lambda_j - \Lambda_l$ should be positive when $l < j$ and negative when $l > j$. Thus the second term in the last line can be positive, making $\tilde{r}_b$ larger than r_b, which means that for a 2-view graph, only enforcing fairness in View 1 can increase the bias in View 2. And this conclusion can be generalized to a multi-view graph, which means that only minimizing the bias of View i (i.e., tr($\mathbf{U}^T \mathbf{L}_{\mathbf{S}_i}\mathbf{U}$)) can increase the bias of View j. ∎

Lemma 1 states that, as long as the eigenvalues of the node similarity matrix satisfy certain condition, it is possible to amplify bias on one view if we enforce individual fairness on another view. In addition to the theoretical analysis, we further conduct empirical evaluations on two datasets (Table II). From the table, we can see that it is indeed possible that when enforcing fairness in one view, the bias of another view is amplified (i.e., $\tilde{r}_{b2} > r_{b2}$).

Table II: Empirical evidence that enforcing individual fairness on View 1 could increase the bias on View 2. $\tilde{r}_{b_2}$ and r_{b_2} are the individual biases of View 2 with and without enforcing individual fairness on View 1, respectively. See the dataset description in Table IV.

Dataset	r_{b2}	$\tilde{r}_{b2}$	$\tilde{r}_{b2}/r_{b2}$
IMDB	0.4394	1.0437	2.3751
ACM	1.1376	5.1020	4.4849

B. iFiG from Multi-Objective Optimization Perspective

Intuitively, we can formulate iFiG problem as a regularized optimization problem as Eq. (3). Straightforward as it is, addressing the well-known dilemma between utility and fairness with Eq. (3) is challenging since it calls for a good choice of regularization parameters $\{\alpha_i\}_{i=1}^M$, which often requires either expert knowledge or exhaustive tuning on them.

To address this challenge, we develop its connection with a multi-objective optimization (MOO) problem

$$\begin{aligned}
\min_{\mathbf{Y}} \quad & \{r_0(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta), \\
& r_1(\mathcal{G}_1, \mathbf{Y}, \theta) \ldots r_M(\mathcal{G}_M, \mathbf{Y}, \theta)\}
\end{aligned} \tag{11}$$

where each loss function is associated with a regularization parameter corresponding to an objective function to be optimized in the MOO problem.[2] Then, different choices of regularization parameters correspond to different solutions to the MOO problem. To this end, we first present the definition of the Pareto front, which indicates the solution of a MOO problem.

Definition 1. *[13] (**Pareo front**) Given a set $\mathcal{Y}$ and a set of view-specific loss functions $\mathbf{r}(\mathbf{Y})$, the set of Pareto front is*

$$\mathcal{P}_{\mathcal{V},\mathcal{Y}} = \{\mathbf{Y} \in \mathcal{Y} : \nexists \mathbf{Y}' \in \mathcal{Y} | \mathbf{Y}' \prec \mathbf{Y}\} \tag{12}$$

where $\mathbf{Y}' \prec \mathbf{Y}$ means $\forall i = 0, ...M+1, r_i(\mathbf{Y}') \prec r(\mathbf{Y})$ Then the corresponding Pareto front risks are denoted as

$$\mathcal{P}_{\mathcal{V},\mathcal{Y}}^{\mathcal{R}} = \{\mathbf{r} \in \mathbb{R}^{M+2} : \exists \mathbf{x} \in \mathcal{P}_{\mathcal{V},\mathcal{Y}}, \mathbf{r} = \mathbf{r}(\mathbf{Y})\} \tag{13}$$

By Definition 1, Geoffrion [14] proves that a Pareto front of the MOO problem in Eq. (11) corresponds to the solution of a linear weighted problem.

Theorem 1. *[14] Given a multi-objective optimization problem*

$$\min_{\mathbf{Y}} \quad \{r_0(\mathbf{Y}), r_1(\mathbf{Y}), \ldots, r_M(\mathbf{Y})\} \tag{14}$$

and a convex set $\mathcal{Y}$ such that $\mathbf{Y} \in \mathcal{Y}$, a set of convex loss functions $\{r_v(\mathbf{Y})\}_{v\in\mathcal{V}}$ with respect to $\mathbf{Y}$ and a set of Pareto front risks $\mathcal{P}_{\mathcal{V},\mathcal{Y}}^{\mathcal{R}}$ defined in Eq. (13)*, if $\mathbf{Y}^*$ is a Pareto optimal solution to MOO problem in Eq.* (14)*, it is a solution to the following linear weighted problem with some choice of α.*

$$\forall \hat{r} \in \mathcal{P}_{\mathcal{V},\mathcal{Y}}^{\mathcal{R}} \, \exists \alpha : \hat{r} = \sum_{i=0}^{M} \alpha_i r_i(\mathbf{Y}) \tag{15}$$

a. The Pareto front is convex: $\forall \boldsymbol{r}, \boldsymbol{r}' \in \mathcal{P}_{\mathcal{V},\mathcal{H}}^{\mathcal{R}}, \lambda \in [0,1], \exists \boldsymbol{r}'' \in \mathcal{P}_{\mathcal{V},\mathcal{H}}^{\mathcal{R}} : \boldsymbol{r}'' \preceq \lambda\boldsymbol{r} + (1-\lambda)\boldsymbol{r}'$

[2] We assume the utility loss $\mathbf{r}(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta)$ is associated with a regularization parameter as well. In Eq. (3), its regularization parameter is 1.

b. Every Pareto solution is a solution to $\forall \hat{r} \in \mathcal{P}^{\mathcal{R}}_{\mathcal{V},\mathcal{H}} \exists \mu : \hat{r} = r(\mu)$

Proof: Omitted. ∎

With Theorem 1, the Pareto front of the multi-objective optimization problem corresponds to the optimal regularization parameters of the following linear weighted problem.

$$\min_{\mathbf{r}} \mathbf{r} = \sum_{i=0}^{M} \alpha_i r_i(\mathbf{Y}) \qquad \text{s.t. } \mathbf{Y} \in \mathcal{Y} \tag{16}$$

Then, to find the best regularization parameters of Eq. (3), we could find the Pareto front of Eq. (11) and the corresponding regularization parameters of a linear weighted problem.

However, the Pareto front is often not unique, meaning that there could be multiple different Pareto fronts of a given multi-objective optimization problem. To find the Pareto front of iFiG that achieves the best trade-off between utility and fairness, we further require that the solution should satisfy the Rawlsian difference principle [15], which asks that it is impossible to make one individual better-off without making at least one another worst-off. Mathematically, it can be formulated as a minimax problem, where the solution should minimize the maximum risk across all views and all loss functions, which is formally defined in Definition 2.

Definition 2. ***(Minimax Pareto fair graph mining result)*** $\mathbf{Y}^*$ *is minimax Pareto fair if it minimizes the worst view-specific loss among all Pareto front results.*

$$\mathbf{Y}^* \in \mathop{argmin}_{\mathbf{Y} \in \mathcal{P}_{\mathcal{V},\mathcal{Y}}} \max_{i \in \mathcal{V}} r_i(\mathbf{Y}) = \mathop{argmin}_{\mathbf{Y} \in \mathcal{P}_{\mathcal{V},\mathcal{Y}}} \|\mathbf{r}(\mathbf{Y})\|_\infty$$

We note that the Pareto fair result $\mathbf{Y}^*$ belongs to a set thus is not unique, neither is the corresponding risk $\mathbf{r}^* = \mathbf{r}(\mathbf{Y}^*)$. To obtain an optimal $\mathbf{Y}^*$, we assume that the convex hypothesis is satisfied. Thus, for every loss $\mathbf{r}^*$, there is a weight vector $\boldsymbol{\alpha}^*$ such that $\mathbf{Y}^*$ is a unique solution for Eq. (16). In this case, we can obtain $\mathbf{Y}^*$ by computing the weighted vector $\boldsymbol{\alpha}^*$.

C. iFiG: Algorithm

To compute $\boldsymbol{\alpha}^*$ that corresponds to the minimax Pareto fair graph mining results of Eq. (11), we follow the general procedures of [13] and present our generic algorithmic framework named iFiG in Algorithm 1. The general procedures of our proposed iFiG framework are as follows. We first initialize the loss function $\mathbf{r}$ with weights: $\boldsymbol{\alpha}$ (step 1), then get the largest loss of all views $\overline{\boldsymbol{r}}$ (step 2). After that, we derive a mask vector to determine which views entail losses that are larger than the maximum current loss (step 3), we generate a new set of weights (step 4), then update loss $\boldsymbol{r}$ (step 5) and parameter K (step 6). If the largest loss in $\boldsymbol{r}$ is smaller than maximum current loss $\overline{\boldsymbol{r}}$ (step 7), meaning the new weights reduce the maximum loss, we can update the outputs (step 8). We repeat the above steps (step 3 - step 8) until convergence.

D. Instantiation #1: Multi-View Spectral Clustering

To instantiate iFiG (Algorithm 1) with multi-view spectral clustering, we first present the following optimization problem by integrating Eq. (1) and Eq. (2)

Algorithm 1: iFiG: Individual Fairness on Multi-View Graph Mining

input : initial weights $\boldsymbol{\alpha}$, adjacency matrices $\{\mathbf{A}_i\}_{i=1}^M$, similarity matrices $\{\mathbf{S}_i\}_{=1}^M$, loss function: $r_i(\cdot), K_{min}$
output: weighted vectors $\boldsymbol{\alpha}^*$, loss $\boldsymbol{r}^*$, graph mining result $\mathbf{Y}^*$

1 **initialize**
2 Initialize graph mining result $\mathbf{Y}$ $\quad \mathbf{Y}, \boldsymbol{r}(\boldsymbol{\alpha}) \leftarrow \mathop{\mathrm{argmin}}_{\mathbf{Y} \in \mathcal{Y}} \sum \alpha_i r_i(\mathbf{Y})$.
3 Get the maximum loss $\overline{\boldsymbol{r}} \leftarrow \|\boldsymbol{r}(\boldsymbol{\alpha})\|_\infty$; $K \leftarrow 1$.
4 **while** *not converged* **do**
5 Compute the mask vector $\quad \mathbf{1}_\alpha \leftarrow \{\mathbb{1}\left(r_i(\boldsymbol{\alpha}) \geq \overline{\boldsymbol{r}}\right)\}_{i=0}^{M+1}$
6 Update weighted vector $\quad \boldsymbol{\alpha} \leftarrow (\mu\boldsymbol{\alpha} + \frac{1-\mu}{K\|\mathbf{1}_\alpha\|_1^1}\mathbf{1}_\alpha)\frac{K}{(K-1)\mu+1}$
7 Update graph mining result and loss $\quad \mathbf{Y}, \boldsymbol{r}(\alpha) \leftarrow \mathop{\mathrm{argmin}}_{\mathbf{Y} \in \mathcal{Y}} \sum \alpha_i r_i(\mathbf{Y})$
8 Update step $K \leftarrow K + 1$
9 **if** $\|\boldsymbol{r}(\alpha)\|_\infty < \overline{\boldsymbol{r}}$ **then**
10 Update the maximum loss $\overline{\boldsymbol{r}} \leftarrow \|\boldsymbol{r}(\alpha)\|_\infty$,
11 Update step $K \leftarrow \min(K, K_{\min})$,
12 Update the results $\mathbf{Y}^*, \boldsymbol{\alpha}^*, \boldsymbol{r}^* \leftarrow \mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{r}(\boldsymbol{\alpha})$
13 **end**
14 **end**
15 **return** *weighted vectors* $\boldsymbol{\alpha}^*$*, loss* $\boldsymbol{r}^*$*, graph mining result* $\mathbf{Y}^*$

$$\min_{\mathbf{Y}} \sum_{i=1}^{M} \mathrm{tr}(\mathbf{Y}^T\mathbf{L}_i\mathbf{Y}) - \lambda \mathrm{tr}(\mathbf{Y}^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{Y}) + \alpha_i \mathrm{tr}\left(\mathbf{Y}^T\mathbf{L}_{\mathbf{S}i}\mathbf{Y}\right) \tag{17}$$

where λ is the weighted parameter balanced coefficient for merging different views. It is intuitive that Eq. (17) is a linear weighted problem

$$r = \sum_{i=0}^{M+1} \alpha_i r_i, \; (\alpha_i > 0 \; i = 0, 1, ..., M+1) \tag{18}$$

where the loss function and the corresponding coefficient are presented in Table III. Then the optimal choice of weight vector $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, ..., \alpha_{M+1}\}$ that balances the utility (r_0 and r_{M+1}) and fairness ($r_i, \forall i = 1, \ldots, M$) can be obtained by finding the Pareto front of the following multi-objective optimization problem using iFiG.

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \{r_0(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta), \\ & r_1(\mathcal{G}_1, \mathbf{Y}, \theta) \ldots r_M(\mathcal{G}_M, \mathbf{Y}, \theta), \, r_{M+1}(\mathcal{G}_M, \mathbf{Y}, \theta)\} \end{aligned} \tag{19}$$

Note that a key step in initiating multi-view spectral clustering with iFiG (Algorithm 1) is step 7, which finds the graph mining results with the updated weight vector. To obtain the mining result, Eq. (17) can be re-written as

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \text{tr}\Big(\mathbf{Y}^T \sum_{i=1}^{M} \big(\mathbf{L}_i + \alpha_i \mathbf{L}_{\mathbf{S}i} - \lambda \mathbf{Y}_i \mathbf{Y}_i^T\big)\mathbf{Y}\Big) \\ \text{s.t.} \quad & \mathbf{Y}^T\mathbf{Y} = I \end{aligned} \tag{20}$$

where $\mathbf{Y} \in \mathbb{R}^{n\times k}$ is the representative subspace (i.e., a soft clustering membership matrix) with k being the number of clusters. From Eq. (20), we observe that $\mathbf{Y}$ is essentially the eigenvectors associated with k smallest eigenvalues of $\sum_{i=1}^{M}\big(\mathbf{L}_i + \alpha_i \mathbf{L}_{\mathbf{S}i} - \lambda \mathbf{Y}_i \mathbf{Y}_i^T\big)$. Thus, debiasing multi-view spectral clustering is spectral clustering on an augmented graph whose graph Laplacian is $\sum_{i=1}^{M}\big(\mathbf{L}_i + \alpha_i \mathbf{L}_{\mathbf{S}i} - \lambda \mathbf{Y}_i \mathbf{Y}_i^T\big)$.

Table III: Notations for individual fairness in multi-view spectral clustering.

View v	Loss Function r	Coefficient α
$v = 0$	$\sum_{i=1}^{M} \text{tr}[\mathbf{Y}_i^T \mathbf{L}_i \mathbf{Y}_i]$	α_0
$v = 1, 2, \ldots, M$	$\mu_s \text{tr}[\mathbf{Y}_i^T \mathbf{L}_{\mathbf{S}i} \mathbf{Y}_i]$	α_v
$v = M+1$	$-\mu_Y \sum_{i=1}^{M} \text{tr}(\mathbf{Y}\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{Y})$	λ

E. Instantiation #2: INFORM *Post-processing*

We present another example by instantiating iFiG (Algorithm 1) with INFORM post-processing [6]. And the instantiation is particularly useful in case the algorithm administrator is not accessible to the mining model or the mining model is too computationally expensive to be re-trained several times (step 7 in Algorithm 1). We first formulate the instantiation as a regularized optimization problem.

$$\min_{\mathbf{Y}} \sum_{i=1}^{M} \|\mathbf{Y} - \hat{\mathbf{Y}}_i\|_F^2 + \alpha_i \text{tr}\left(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}i} \mathbf{Y}\right) \tag{21}$$

where $\hat{\mathbf{Y}}_i$ denotes the vanilla mining results of i-th view. Then Eq. (21) can be represented as a linear weighted problem

$$r = \sum_{i=0}^{M} \alpha_i r_i, \; (\alpha_i > 0 \, i = 0, 1, ..., M) \tag{22}$$

where $r_0 = \sum_{i=1}^{M} \|\mathbf{Y} - \hat{\mathbf{Y}}_i\|_F^2$ and $r_i = \text{tr}\left(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}i} \mathbf{Y}\right), i = 1, ..., M$. Then we can instantiate iFiG and solve it from the perspective of multi-objective optimization (MOO).

$$\min_{\mathbf{Y}} \quad \{r_0(\mathcal{G}^{\text{multi}}, \mathbf{Y}, \theta), r_1(\mathcal{G}_1, \mathbf{Y}, \theta) \, \ldots \, r_M(\mathcal{G}_M, \mathbf{Y}, \theta)\} \tag{23}$$

Then, to update the mining results $\mathbf{Y}$ and the corresponding risks with the updated weight vector (step 7), following [6], we could efficiently find its closed-form solution by solving the linear system $\sum_{i=1}^{M}(\mathbf{I} + \alpha_i \mathbf{L}_{\mathbf{S}_i})\mathbf{Y} = \sum_{i=1}^{M} \hat{\mathbf{Y}}_i$ with any linear system solver (e.g., conjugate gradient method).

IV. EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed iFiG algorithm to answer the following questions.

RQ1. How effective is iFiG in multi-view graph clustering?
RQ2. How effective is iFiG in enforcing individual fairness?

A. Experimental Setup

Datasets. We test the proposed method on five datasets, including one synthetic dataset and four real-world graphs. All real-world graphs are publicly available. The statistics of these datasets are summarized in Table IV. The detailed descriptions of these datasets are as follows.

Table IV: Statistics of datasets.

Dataset	# Views	# Nodes	# Edges	# Clusters
Synthetic	2	1000	254072 243074	2
DBLP	3	7,800	6,106 855 108,896	4
IMDB	2	3,550	66,428 13,778	3
Twitter	2	2,000	1,797 1,152	N/A
ACM	2	3,025	1,106,893 16,153	3

- *Synthetic* consists of two views following the same data generation strategy and parameter settings in [1]. We refer to [1] for more details.
- *Twitter* [3] is one of the biggest online social platforms. The Twitter dataset [16] contains two views, where nodes are users, and the edges represent the reply network and the mentioned network, respectively. Here we sample 2,000 nodes from the dataset.
- *IMDB* [4] is an online database containing information about movies, actors, and directors. The IMDB dataset [17] consists of two views, whose nodes represent movies, edges represent movie-actor-movie and movie-director-movie relationships, respectively.
- *DBLP* [5] is a computer science bibliography website for open bibliographic information. The DBLP dataset [18] is a multi-view graph whose nodes are papers. The dataset contains three views corresponding to paper-paper citation relationship, paper-author-paper relationship and paper-author-term-author-paper relationship, respectively.
- *ACM* [6] is a dataset extracted by Wang et al. [17]. It contains 3025 papers published in KDD, SIGMOD, SIGCOMM, MobiCOMM, and VLDB. There are 2 views in the dataset whose edges are paper-author-paper relationship and paper-subject-paper relationship, respectively.

Baseline Methods. We compare iFiG (Algorithm 1) with several baseline methods. Each baseline method is briefly summarized as follows.

- *Spectral clustering (SC)* [19] finds the soft cluster membership matrix of each node in a single-view graph by

[3] https://twitter.com/
[4] https://www.imdb.com/
[5] https://dblp.org/
[6] http://dl.acm.org/

analyzing the spectrum of its graph Laplacian. From an optimization perspective, it is often solved by finding the eigenvectors associated with the k smallest eigenvalues where k is the number of clusters to find. In our experiments, following the strategy in [3], we first preprocess a flattened adjacency matrix $\tilde{\mathbf{A}}$ by getting the union of the adjacency matrices of different views ($\tilde{\mathbf{A}}[i,j]=0$ if there is no edge between node i and node j in any view, and 1 otherwise) and then perform spectral clustering on the flattened graph.

- *SC-ML* [10] applies spectral clustering to multi-layer graphs using a dimensionality reduction framework. It first considers each layer as a subspace and combines the representations of all subspaces into a new target subspace. Then, it is formulated as an optimization problem (Eq. (1)) and can be solved by performing spectral clustering on a modified graph Laplacian.
- *InFoRM* [6] ensures individual fairness on mining plain graph through Laplacian regularization on the pairwise node similarity matrix of the graph (Eq. (2)). We first adopt the same preprocessing strategy as *SC* to get the flattened graph and then compute the node-node similarity matrix using Jaccard similarity. The flattened graph and the similarity matrix serve as the input of InFoRM algorithm.
- *Gen-FairSC* [20] studies individually fair spectral clustering on a single-view graph. It leverages a representation graph and requires the neighbors of a node in the representation graph to be approximately proportionally represented in the clusters. In our experiments, we use the same pairwise node similarity matrix in InFoRM as the representation graph.

Evaluation Metrics. To answer **RQ1**, we measure the clustering quality with three different metrics including accuracy (Acc.), F1 score (F1) and normalized mutual information (NMI). More specifically, for the first four datasets, we compare the quality of predicted cluster assignment with the ground truth using accuracy (Acc.), F1 score (F1) and NMI score (NMI). Since we don't have the ground truth for the Twitter dataset, we evaluate the models by comparing the NMI score between SC-ML and other methods. To answer **RQ2**, we use $Reduction = 1 - \frac{\mathrm{tr}((\mathbf{Y}^*)^T \mathbf{L_S} \mathbf{Y}^*)}{\mathrm{tr}(\bar{\mathbf{Y}}^T \mathbf{L_S} \bar{\mathbf{Y}})}$ to measure the relative reduction of individual bias where $\bar{\mathbf{Y}}$ is the soft cluster membership matrix obtained by SC-ML and $\mathbf{Y}^*$ is the soft cluster membership matrix obtained by the baseline methods or our proposed method. In the definition of relative reduction *Reduction*, the numerator is the individual bias of the vanilla SC-ML, while the denominator is the individual bias of the baseline methods or our proposed method. Thus, it measures to what extent the individual bias is reduced concerning SC-ML without fairness considerations.

Parameter Settings. For the synthetic data, the weighted parameter α is 0.6 in *SC-ML* and 10000 in *InFoRM*. For the DBLP dataset, the weighted parameter α is 0.5 in *SC-ML* and 10000 in *InFoRM*. For the IMDB dataset, the weighted parameter α is 0.7 in *SC-ML* and 0.5 in *InFoRM*. For the Twitter dataset, the weighted parameter α is 0.5 in *SC-ML* and 0.5 in *InFoRM*, we set the number of clusters to be 5 for a relatively better clustering performance. For the ACM dataset, the weighted parameter α is 0.6 in *SC-ML* and 10000 in *InFoRM*. For iFiG the initial weights are 0.25 in the synthetic dataset, IMDB dataset, Twitter dataset, and the ACM dataset and 0.2 in the DBLP dataset.

Machine Configuration and Reproducibility. All codes are written in Python 3.8, NumPy 1.20 and NetworkX 2.5. All experiments are performed on a Linux server with 96 Intel Xeon Gold 6240R CPUs. We will release the source code of the proposed methods as well as the synthetic data upon the publication of the paper.

Table V: Effectiveness results on multi-view graph clustering. Higher is better.

Dataset	Method	Acc.	F1	NMI	Reduction
Synthetic	SC	50.3000	0.3371	0.0014	0.0040
	SC-ML	50.1000	0.3344	0.0020	0
	InFoRM	67.4000	0.6643	0.1381	0.1068
	Gen-FairSC	49.9500	0.3331	0.0096	0.5731
	iFiG (Ours)	52.2000	0.3329	0.0223	0.6335
DBLP	SC	30.6364	0.1181	0.0015	0.9979
	SC-ML	30.6194	0.1172	0.0009	0
	InFoRM	30.6025	0.1171	0.0012	0.9982
	Gen-FairSC	30.6533	0.1175	0.0009	0.6137
	iFiG (Ours)	30.6364	0.1175	0.0012	0.9984
IMDB	SC	38.1408	0.2009	0.0034	0.9999
	SC-ML	37.6056	0.1997	0.0040	0
	InFoRM	38.0563	0.1937	0.0024	1.0000
	Gen-FairSC	37.7746	0.1839	0.0012	-472.2874
	iFiG (Ours)	37.9155	0.2158	0.0055	1.0000
ACM	SC	35.0744	0.1738	0.0011	1.0000
	SC-ML	35.0744	0.1739	0.0025	0
	InFoRM	35.0413	0.1730	0.0013	1.0000
	Gen-FairSC	35.0082	0.1729	0.0013	-110.6294
	iFiG (Ours)	34.4132	0.2085	0.0073	0.9999
Twitter	SC	-	-	0.2717	0.7662
	SC-ML	-	-	1	0
	InFoRM	-	-	0.2168	0.7958
	Gen-FairSC	-	-	0.1605	1.0000
	iFiG (Ours)	-	-	0.2896	0.8835

B. Main Result

Table V presents the quantitative results of iFiG and the baseline methods on all datasets. Since *Twitter* does not have ground truth for cluster assignment, we do not report the accuracy (Acc.) and F1 score (F1) for all compared methods. From the table, we observe that iFiG consistently mitigates the bias (i.e., Reduction) without sacrificing too much accuracy (i.e., Acc., F1 and NMI). We note that, due to different magnitudes of reductions for different methods, the reduction for *IMDB* and *ACM* are rounded up to 1. For *ACM*, we can reduce the bias while preserving the performance in terms of F1-score and NMI score. For the *Twitter* dataset, though Gen-

Table VI: Effectiveness results on INFORM post-processing. Higher is better.

Dataset	Method	Acc.	F1	NMI	Reduction
Synthetic	SC	50.9000	0.3628	0.0052	0.0042
	SC-ML	50.1000	0.3356	0.0019	0
	InFoRM	50.0000	0.3351	0.0052	0.9999
	Gen-FairSC	49.3000	0.3302	0.0052	0.4935
	iFiG (Ours)	50.2000	0.3356	0.0024	0.9999
DBLP	SC	28.7087	0.1119	0.0018	0.0091
	SC-ML	30.2538	0.1129	0.0014	0
	InFoRM	30.2538	0.1129	0.0014	0.9999
	Gen-FairSC	30.3871	0.1116	0.0046	-164.5330
	iFiG (Ours)	30.6125	0.1135	0.0018	0.8420
IMDB	SC	37.8028	0.2024	0.0014	1.0000
	SC-ML	37.6901	0.1987	0.0020	0
	InFoRM	37.4647	0.1915	0.0027	0.9999
	Gen-FairSC	37.7183	0.1827	0.0012	-7.1781
	iFiG (Ours)	38.1408	0.2127	0.0037	0.9999
ACM	SC	35.1074	0.1741	0.0032	1.0000
	SC-ML	30.1074	0.1741	0.0041	0
	InFoRM	35.0413	0.1730	0.0013	0.9999
	Gen-FairSC	35.0413	0.1730	0.0013	-52.6667
	iFiG (Ours)	33.2893	0.1970	0.0664	0.9999
Twitter	SC	-	-	0.1777	0.9819
	SC-ML	-	-	1	0
	InFoRM	-	-	0.1671	0.9999
	Gen-FairSC	-	-	0.0022	-121.6472
	iFiG (Ours)	-	-	0.1793	0.9991

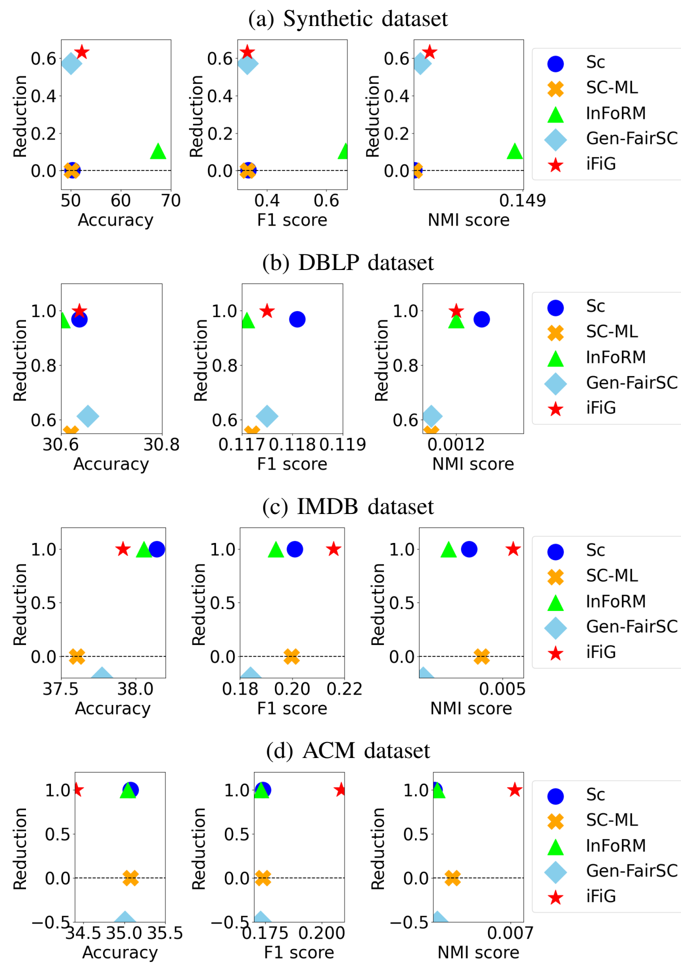

Figure 2: Trade-off between Accuracy/F1 and Reduction. Best viewed in color. Red star represents iFiG. The dashed line in each figure represents the reduction score of SC-ML. The closer to upper right, the better trade-off between utility (i.e., Accuracy/F1) and fairness (i.e., Reduction).

FairSC mitigates more bias than iFiG, its NMI is severely reduced compared with all other methods.

In addition, the trade-off between accuracy/F1 score and reduction results are shown in Figure 2. From the figure, we observe that iFiG can reduce bias and at the same time without losing too much accuracy in most cases.

Evaluation results for post-processing are also shown in Table VI. From these tables, we can see that our proposed methods can effectively enforce fairness and at the same time preserve the performance of spectral clustering.

C. Pareto Optimal Analysis

1) Weight analysis: Other than the performance on debiasing, we analyze how the weight parameters are related to loss functions. Following [21], given a Pareto optimal solution $\mathbf{r}$ with a weight vector α, any two loss functions $r_1, r_2 \in \mathbf{r}$ and their weights α_1, α_2 satisfy $\frac{dr_2}{dr_1} = -\frac{\alpha_1}{\alpha_2}$ where the left hand side can be represented as the trade-off between the risk of View 1 (r_1) and the risk of View 2 (r_2). Considering the weight of utility loss α_0 and the weights of individual fairness of each view $\alpha_i, (i = 1, ..., M)$, the right hand side can be interpreted as how enforcing individual fairness $\text{tr}(\mathbf{U}^T \mathbf{L}_{\mathbf{S}_i} \mathbf{U})$ on each view can lead to an increase in the utility loss. The slope $-\frac{\alpha_i}{\alpha_0}$ $(i = 1, 2, ..., M)$ for each dataset is shown in Table VII. We find that in the synthetic dataset, the slopes of the two views are the same, possibly for the reason that the data points in two views are generated similarly. We also observe that in the IMDB dataset, slopes of two views are approximately -1 while in the DBLP dataset, slopes are approximately -1 for View 1 and 3 but smaller than 1 for View 2. Since the slopes in the DBLP dataset are smaller than those in the IMDB dataset, the trade-off of enforcing fairness in the DBLP dataset is smaller.

2) Convergence analysis: We also analyze how losses in different views $\{r_i\}_{i=0}^{M+1}$ change during iterations. From Figure 3, we observe that the view-specific losses converge to their final values.

Table VII: Slope of each view.

Dataset	View 1	View 2	View 3
Synthetic	-1.2925	-1.2925	-
DBLP	-0.4918	-0.4918	-0.3278
IMDB	-1.7000	-4.1000	-
Twitter	-1.0000	-5.0000	-
ACM	-0.1546	-0.1150	-

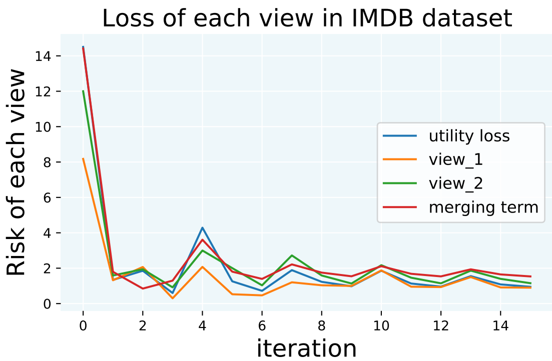

Figure 3: An illustration of how losses change for each iteration. The merging term is $\sum_{i=1}^{M} \mathrm{tr}(\mathbf{U}\mathbf{U}_i\mathbf{U}_i^T\mathbf{U})$

V. Related Work

In this section, we review related work from the perspectives of (1) multi-view graph mining and (2) fair graph mining.

A - Multi-view graph mining aims to explore real-world networks where nodes are usually connected by different relations for improving the performance of mining. As multiple graph mining models have been proposed, there are also many multi-view graph mining methods, the most common of which is clustering. For clustering, Kumar et al. [1] propose a multi-view spectral clustering approach that leverages co-regularization to minimize the disagreement among clusters from different views. Dong et al. [10] propose to combine all views of a graph together with distance analysis on a Grassmann manifold, then use the traditional spectral clustering method. Brbic and Kopriva [22] maximize the agreement across different views by learning a joint subspace representation across all views. Liang et al. [23] learn a unified graph by preserving the view consistent parts while at the same time removing the inconsistent parts. Pan and Kang [24] propose a contrastive learning based clustering method for multi-view attributed graphs considering noisy and incomplete data. Wang et al. [25] propose to obtain weights of different views automatically to unify all views. They also construct the graph of each view and the unified graph jointly and produce the final clusters without additional clustering steps. Fan et al. [26] consider clustering on attributed multi-view graphs and develop a graph autoencoder-based clustering framework to model the multi-view graph information. Regarding representation learning, Qu et al. [27] propose an attention-based multi-view representation learning approach that weights each view differently and focuses on the most informative views for more robust representations across different views. Shi et al. [2] learn multi-view graph embeddings by modeling the view consistency (i.e., collaboration) and the view-specific semantics (i.e., preservation) simultaneously. Cen et al. [28] consider the embeddings for attributed multiplex heterogeneous networks and propose a unified framework that supports both transductive and inductive learning. Hassani and Khasahmadi [29] leverage self-supervised learning to learn the node-level and graph-level representations with mutual information maximization in a contrastive manner. Fu et al. [30] propose a view-adversarial framework to generate multi-view network representations. Jing et al. [31] propose HDMI that learns node embeddings on multi-view graphs considering intrinsic self-supervised signal. Khan and Blumenstock [3] first merge different views using subspace analysis, then apply a graph convolutional network for semi-supervised node classification tasks.

B – Fair graph mining seeks to mitigate bias in graph mining models, which is an emerging research topic. For group fairness on graphs, a vast majority of existing works consider single-view graphs only. Tsioutsiouliklis et al. [32] ensure group fairness in PageRank [33]. Rahman et al. [34] extends node2vec [35] with a fair random walk to ensure statistical parity. Kleindessner et al. [5] divide nodes into clusters so that every group is approximately proportionally represented in each cluster. Our work differs from [5] in that we aim to ensure individual fairness in clustering a multi-view graph, whereas [5] considers group fairness on spectral clustering for a single-view graph. Dai and Wang et al. [36] propose an adversarial learning based framework to alleviate the bias in graph neural networks (GNN) with limited sensitive information. For the few works that consider graphs with multiple edge/node types, Bose and Hamilton [8] propose an adversarial learning based framework to learn fair graph embeddings for a collection of sensitive attributes. Nevertheless, Bose and Hamilton [8] consider a different type of fairness notion (i.e., group fairness) from our work (i.e., individual fairness). Zeng et al. [9] ensure group fairness in heterogeneous network embedding with a variety of bias mitigation strategies, including sampling-based, projection-based, and GNN-based techniques. Different from Zeng et al. [9], we consider individual fairness on multi-view graphs instead of group fairness on heterogeneous networks. Regarding individual fairness, Kang et al. [6] provide the first principled study for individual fairness on single-view graph mining. Dong et al. [37] propose a ranking-based individual fairness definition to avoid direct distance comparison. However, none of the existing works [6], [37] considers individual fairness on multi-view graphs. There are also other types of fairness considered in graph mining. For example, through contrastive learning, Agarwal et al. [7] learn counterfactually fair graph embedding. Ma et al. [38] use Siamese networks [39] to ensure counterfactual fairness in graph embeddings. Kang et al. [40] mitigate degree-related bias in graph convolutional networks by analyzing the gradient computation. Rahmattalabi et al. [41] propose a fair and robust graph covering that ensures a certain proportion of nodes in each demographic group are included in the mining results.

VI. Conclusion

In this paper, we study individual fairness in multi-view graph mining (iFiG problem). We first provide both theoretical and empirical analysis on the necessity of enforcing individual fairness on multi-view graphs. Then we formulate the problem as a linear weighted optimization problem and resort to the minimax Pareto fairness, whose rationality is rooted in the

Rawlsian difference principle. Based on that, we propose an effective solver named iFiG that minimizes the utility loss while promoting individual fairness for each view. Moreover, we conduct extensive experiments on diverse synthetic and real-world datasets, demonstrating that the proposed method is effective in mitigating the individual bias in a multi-view graph while largely maintaining the performance of various graph mining tasks.

ACKNOWLEDGMENT

This work is supported by NSF (1947135, and 2134079), the NSF Program on Fairness in AI in collaboration with Amazon (1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799), and ARO (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," *Advances in neural information processing systems*, vol. 24, pp. 1413–1421, 2011.

[2] Y. Shi, F. Han, X. He, X. He, C. Yang, J. Luo, and J. Han, "mvn2vec: Preservation and collaboration in multi-view network embedding," *arXiv preprint arXiv:1801.06597*, 2018.

[3] M. R. Khan and J. E. Blumenstock, "Multi-gcn: Graph convolutional networks for multi-view networks, with applications to global poverty," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 606–613.

[4] Y. Li, Y. Ning, R. Liu, Y. Wu, and W. Hui Wang, "Fairness of classification using users' social relationships in online peer-to-peer lending," in *Companion Proceedings of the Web Conference 2020*, 2020.

[5] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern, "Guarantees for spectral clustering with fairness constraints," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3458–3467.

[6] J. Kang, J. He, R. Maciejewski, and H. Tong, "Inform: Individual fairness on graph mining," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 379–389.

[7] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," *arXiv preprint arXiv:2102.13186*, 2021.

[8] A. Bose and W. Hamilton, "Compositional fairness constraints for graph embeddings," in *International Conference on Machine Learning*. PMLR, 2019, pp. 715–724.

[9] Z. Zeng, R. Islam, K. N. Keya, J. Foulds, Y. Song, and S. Pan, "Fair representation learning for heterogeneous information networks," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, 2021, pp. 877–887.

[10] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on signal processing*, vol. 62, no. 4, 2013.

[11] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[12] G. W. Stewart, "Matrix perturbation theory," 1990.

[13] N. Martinez, M. Bertran, and G. Sapiro, "Minimax pareto fairness: A multi objective perspective," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6755–6764.

[14] A. M. Geoffrion, "Proper efficiency and the theory of vector maximization," *Journal of mathematical analysis and applications*, vol. 22, no. 3, pp. 618–630, 1968.

[15] J. Rawls, "A theory of justice," in *A theory of justice*. Harvard university press, 2020.

[16] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[17] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[18] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 990–998.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[20] S. Gupta and A. Dukkipati, "Protecting individual interests across clusters: Spectral clustering with guarantees," *arXiv preprint arXiv:2105.03714*, 2021.

[21] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[22] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognition*, vol. 73, pp. 247–258, 2018.

[23] Y. Liang, D. Huang, C.-D. Wang, and P. S. Yu, "Multi-view graph learning by joint modeling of consistency and inconsistency," *arXiv preprint arXiv:2008.10208*, 2020.

[24] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[25] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.

[26] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, "One2multi graph autoencoder for multi-view graph clustering," in *Proceedings of The Web Conference 2020*, 2020, pp. 3070–3076.

[27] M. Qu, J. Tang, J. Shang, X. Ren, M. Zhang, and J. Han, "An attention-based collaboration framework for multi-view network representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1767–1776.

[28] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1358–1368.

[29] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 4116–4126.

[30] D. Fu, Z. Xu, B. Li, H. Tong, and J. He, "A view-adversarial framework for multi-view network embedding," in *The 29th ACM International Conference on Information and Knowledge Management*, 2020, pp. 2025–2028.

[31] B. Jing, C. Park, and H. Tong, "Hdmi: High-order deep multiplex infomax," in *Proceedings of the Web Conference 2021*, 2021.

[32] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, and N. Mamoulis, "Fairness-aware pagerank," in *Proceedings of the Web Conference 2021*, 2021, pp. 3815–3826.

[33] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[34] T. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards fair graph embedding," 2019.

[35] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[36] E. Dai and S. Wang, "Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 680–688.

[37] Y. Dong, J. Kang, H. Tong, and J. Li, "Individual fairness for graph neural networks: A ranking based approach," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 300–310.

[38] J. Ma, R. Guo, M. Wan, L. Yang, A. Zhang, and J. Li, "Learning fair node representations with graph counterfactual fairness," *arXiv preprint arXiv:2201.03662*, 2022.

[39] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[40] J. Kang, Y. Zhu, Y. Xia, J. Luo, and H. Tong, “Rawlsgcn: Towards rawlsian difference principle on graph convolutional network,” in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1214–1225.

[41] A. Rahmattalabi, P. Vayanos, A. Fulginiti, E. Rice, B. Wilder, A. Yadav, and M. Tambe, “Exploring algorithmic fairness in robust graph covering problems,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 15 776–15 787, 2019.